Interdisciplinary Web Design

DEPOT BUILDING - Rm. 2002
814 W. Broad Street

Andrew Ilnicki – ilnickiap@vcu.edu
Software

Adobe Creative Suite, including Dreamweaver

Text Editor - Coda, Komodo Edit, Espresso

Personal Web Server

FTP Client
Books

HTML5 For Web Designers
By Jeremy Keith

Handcrafted CSS
By Dan Cederholm

[other references will be distributed in class]
You Will Learn

- HTML / CSS Best Practices
- JavaScript / jQuery Best Practices
- PHP - dynamic web development
- CMS - content management systems (Wordpress, Indexhibit)
You Will Try

CSS3

Responsive Web Design

Custom CMS Development with Wordpress for a real client

SEO - search engine optimization
You Will Benefit From

Practical, Transferable, Real world skills + experience

Responsive Web Design implementation

Professional Practices, tips, and tricks

Your new found hire-ability as a hybrid designer / web developer
XHTML PRIMER

Day 1 intro
What is XHTML?
What is XHTML?

• XHTML means that when developing webpages, you must keep your presentation...

---------------------

separate from...

---------------------

• your content
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel="stylesheet" type="text/css" media="all" href="styles.css" />
<script type="text/javascript" src="behaviors.js"></script>
</head>

<body>
This is where the content goes....
</body>

</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel="stylesheet" type="text/css" media="all" href="styles.css" />
<script type="text/javascript" src="behaviors.js"></script>
</head>
<body>
This is where the content goes....
</body>
</html>
Why Use XHTML?

• XHTML adherence encourages good practice.

• All your markup will be well-formed and all your tags will be closed.

• Makes page rendering easier for browsers

• Makes bug-tracking easier for you.

• Web pages will be easier to edit, particularly in terms of site-wide changes.
Layers within Layers

- Content Layer
- Presentation Layer
- Behavior Layer
Layers within Layers

* Content Layer
  Text and Images

* Presentation Layer
  Styles – color, composition, typography

* Behavior Layer
  Interaction – Javascript, Actionscript/Flash
But, here’s the catch...

• **XHTML is now considered a series of “Best Practices”**

  • it is too stringent, too complicated, and too limiting to enforce sweepingly across the WWW.

  • However, the philosophy and concept of XHTML separation lives on as a basis for ...
HTML5

- **HTML5 is relatively new**
  - not all browsers support HTML5 *(Internet Explorer 6 + 7)*
  - ...but eventually they will

- it’s a simpler syntax to learn and memorize, so feel lucky!

- intended to force the browser to handle commonly used solutions like video and audio innately... instead of using plugins.
HTML5 Example

•  doctype

OLD

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

NEW

<!DOCTYPE html>
HTML5 Example

- scripts, styles, and types

OLD

```html
<link rel="stylesheet" href="path/to/stylesheet.css" type="text/css" />
<script type="text/javascript" src="path/to/script.js"></script>
```

NEW

```html
<link rel="stylesheet" href="path/to/stylesheet.css" />
<script src="path/to/script.js"></script>
```
HTML5 Example

- Video embedding

OLD

<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" width="100" height="100" codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,40,0"><param name="allowFullScreen" value="true" /><param name="allowscriptaccess" value="always" /><param name="src" value="http://www.youtube.com/v/Sv5iEK-IEzw?fs=1&hl=en_GB" /><param name="allowfullscreen" value="true" /><embed type="application/x-shockwave-flash" width="100" height="100" src="http://www.youtube.com/v/Sv5iEK-IEzw?fs=1&hl=en_GB" allowscriptaccess="always" allowfullscreen="true"></embed></object>

NEW

<video src=myvideo.mp4 width=640 height=480 controls poster=myImage.jpg></video>
My hypothesis

HTML5 Improves:
Learning and retention of web design and development best practices
Primer - CSS

• CSS – Cascading Stylesheets

• example:

```html
html {
    margin:0;
    padding:0;
}
```
Why Use CSS?

➡ just imagine all the benefits that come with separating your presentation from your content.
SAY YES TO CSS!

• Your pages will be smaller. Much, much, much smaller.

  • Without the bloat that comes with nested tables, spacer images and font tags, your mark-up will be leaner and meaner.

  • That will appeal to search engines.
SAY YES TO CSS!

• Life will be simpler for the people in charge of the design

The presentation of an entire site full of documents can be changed by altering just one file without ever touching the content.
SAY YES TO CSS!

• Life will also be simpler for the people in charge of the content.

Your mark-up will be human readable allowing the content to be updated without changing the rules that govern the presentation.
SAY YES TO CSS!

• Your content will be accessible to all browsing devices, past and present

That means web-enabled mobile devices too!
A DETAILED EXAMPLE
(re-define the defaults)
@charset "UTF-8";

/* CSS Document */

* { padding: 0; margin: 0; }

html, body, div, span, applet, object, iframe, h1, h2, h3, h4, h5, h6, p, blockquote, pre, a, abbr, acronym, address, big, cite, code, del, dfn, em, font, img, ins, kbd, q, s, samp, small, strike, strong, sub, sup, tt, var, dl, dt, dd, ol, ul, li, fieldset, form, label, legend, table, caption, tbody, tfoot, thead, tr, th, td {
    margin: 0;
    padding: 0;
    border: 0;
    outline: 0;
    font-size: 100%;
    vertical-align: baseline;
}

/* remember to define focus styles! */
:focus {
    outline: 0;
}

body {
    line-height: 1;
    color: black;
    background: white;
}

ol, ul {
    list-style: none;
}

/* tables still need 'cellspacing="0"' in the markup */
table {
    border-collapse: separate;
    border-spacing: 0;
}

caption, th, td {
    text-align: left;
    font-weight: normal;
}

blockquote:before, blockquote:after, q:before, q:after {
    content: "";
}

blockquote, q {
    quotes: "" "";
}
/* CSS Document */

/* {padding:0; margin:0;} */

html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, font, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td {
    margin: 0;
    padding: 0;
    border: 0;
    outline: 0;
    font-size: 100%;
    vertical-align: baseline;
}

/* remember to define focus styles! */
:focus {
  outline: 0;
}

body {
    line-height: 1;
    color: black;
    background: white;
}

ol, ul {
    list-style: none;
}

/* tables still need 'cellspacing="0"' in the markup */
table {
    border-collapse: separate;
    border-spacing: 0;
}
caption, th, td {
    text-align: left;
    font-weight: normal;
}

blockquote:before, blockquote:after,
q:before, q:after {
    content: "";
}

blockquote, q {
    quotes: "" "";
}
@charset "UTF-8";
/* CSS Document */

* { padding: 0; margin: 0; }

html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, font, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td {
  margin: 0;
  padding: 0;
  border: 0;
  outline: 0;
  font-size: 100%;
  vertical-align: baseline;
}

/* remember to define focus styles! */
:focus {
  outline: 0;
}

body {
  line-height: 1;
  color: black;
  background: white;
}

ol, ul {
  list-style: none;
}

/* tables still need 'cellspacing=0' in the markup */

table {
  border-collapse: separate;
  border-spacing: 0;
}

caption, th, td {
  text-align: left;
  font-weight: normal;
}

blockquote, q {
  quotes: "" "";
}
reset.css

Rules or Styles

@charset "UTF-8";
/* CSS Document */

* {padding: 0; margin: 0;}

html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, font, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td {
  margin: 0;
  padding: 0;
  border: 0;
  outline: 0;
  font-size: 100%;
  vertical-align: baseline;
}

/* remember to define focus styles! */
:focus {
  outline: 0;
}

body {
  line-height: 1;
  color: black;
  background: white;
}

ol, ul {
  list-style: none;
}

/* tables still need 'cellspacing="0"' in the markup */
table {
  border-collapse: separate;
  border-spacing: 0;
}
caption, th, td {
  text-align: left;
  font-weight: normal;
}

blockquote: before, blockquote: after,
q: before, q:after {
  content: "";
}

blockquote, q {
  quotes: "";"
reset.css

```css
@charset "UTF-8";
/* CSS Document */

* { padding: 0; margin: 0; }
html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, font, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td {
  margin: 0;
  padding: 0;
  border: 0;
  outline: 0;
  font-size: 100%;
  vertical-align: baseline;
}
/* remember to define focus styles! */
:focus {
  outline: 0;
}
body {
  line-height: 1;
  color: black;
  background: white;
}
ol, ul {
  list-style: none;
}
/* tables still need 'cellspacing="0"' in the markup */
table {
  border-collapse: separate;
  border-spacing: 0;
}
caption, th, td {
  text-align: left;
  font-weight: normal;
}
blockquote:before, blockquote:after,
q:before, q:after {
  content: "";
}
blockquote, q {
  quotes: "" ;
}
```
@charset "UTF-8";
/* CSS Document */

* { padding: 0; margin: 0; }
html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, font, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td {
    margin: 0;
    padding: 0;
    border: 0;
    outline: 0;
    font-size: 100%;
    vertical-align: baseline;
}

/* remember to define focus styles! */
:focus {
    outline: 0;
}
body {
    line-height: 1;
    color: black;
    background: white;
}
ol, ul {
    list-style: none;
}
/* tables still need 'cellspacing="0"' in the markup */
table {
    border-collapse: separate;
    border-spacing: 0;
}
caption, th, td {
    text-align: left;
    font-weight: normal;
}
blockquote:before, blockquote:after,
q:before, q:after {
    content: "";
}
blockquote, q {
    quotes: "" "";
}
SEMANTICS of code

• use `<h*>` tags
  Put headings and subheadings into `<h>` tags, or `<h1>`, `<h2>`, `<h3>`, etc.

  (HINT)

```css
h1 { font-family: sans-serif }
h2 { font-family: sans-serif }
h3 { font-family: sans-serif }
```

is equivalent to...

```css
h1, h2, h3 { font-family: sans-serif }
```
You can assign style information to all elements with a class of "pastoral" as follows:

```
.pastoral { color: green } /* all elements with class~="pastoral" */
```

The following assigns style only to H1 elements with class of "pastoral":

```
h1.pastoral { color: green } /* H1 elements with class~="pastoral" */
```

[ Given these rules, the first H1 instance in the example code would not have green text, while the second would ]
SEMANTICS of code

• use ‘graphs + lists

Wrap your paragraphs in `<p>` tags. If you have a list of links, then put them in a list (`<ul>` or `<ol>`) and make each of them a list item, `<li>`.

EXAMPLE CODE:

```html
<p>This is a Paragraph</p>

<ol>
  <li>This is an ORDERED list</li>
</ol>

<ul id="styleName">
  <li>This is an UNORDERED list</li>
</ul>
```
DON'T WORRY!

• The default presentation of a tag is ONLY a result of browsers following rules...

CSS CHANGES ALL THAT
NOW TO THE LAYOUT

- `<div>` and `<span>`

EXAMPLE CODE:

```
<div>
    <p>This is a Paragraph inside a DIV element</p>
    <span>
        This is a specific content set that needs to be treated differently inline
    </span>
    <p>This is another Paragraph after the span</p>
</div>
```
NOW TO THE LAYOUT

A `<div>` is a block level element. That means it’s self-contained and comes with a built-in line break.

A `<span>` is an inline element. Inline elements don’t include a line break and must be contained within a block level element.

Both are like blank slates just waiting for styles to be applied to them.
Classes and IDs

body, table, th, td, p, ul, li...

etc.

If you want to apply a style to all instances of a certain tag, then you simply reference that tag in your stylesheet.

```css
body, table, th, td {
  margin: 0;
  padding: 0;
}

p {
  margin-bottom: 18px;
}
```
Be Selective

Classes are intended to be reusable...

<p class="myclass">

but an ID is unique...

<body id="myid">

So an HTML document would only contain one instance of <body id="myid">
Most web pages can be divided up into sections like "main navigation", "sub navigation", "main content" "related material", etc.

XHTML does not give us the tags to describe these chunks of content. But if we take the blank slate block-level tag (<div>) and give each section a unique identifier (id=""), then we've taken semantic description to the next level.

So if you mark up your pages with areas like:

`<div id="branding"`, or
`<div id="footer">`

then you have a way to reference those chunks of content. You will then be able to affect their visual appearance and even change how they are positioned.
The same rules apply to classes, and other elements as well:

- `<div class="specialHeading"`, or
- `<h1 class="redhead">`, or
- `<p class="big-red-text">`, or
- `<span class="inline-pink-notation">`, or
- `<ul class="bulleted-list">`, or
- `<br class="clear">`...
HTML5 Simplifies this

New content areas include:

```html
<header>
<nav>
<footer>
<aside>
<article>
<section>
```
All for today, thank you!